

文章编号: 1005-0523(2005)04-0098-04

基于 ActiveX 控件技术的组态软件图形界面元素的设计

张红斌, 李广丽

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要:对组态软件用户图形界面进行了介绍, 引入了组成图形界面的基本元素—ActiveX 控件, 并详细地分析了三个 ActiveX 控件的设计流程及关键技术.

关键词:ActiveX 控件; 组态软件; VC++ 6.0; MFC

中图分类号:TP393

文献标识码:A

1 组态软件简介

组态软件的使用和开发已经有了几十年的历史, 早期的组态软件多是基于 UNIX 和 DOS, 近年, 随着 Windows 的普及, 越来越多的组态软件开始选择 Windows 作为其开发平台. 在 Windows 平台上一套完整的组态软件包括四个部分: 应用程序管理器、用户图形界面、实时数据库和 I/O 驱动程序.

用户图形界面(Graphic User Interface 下简称 GUI)在组态软件的使用过程中起着举足轻重的作用, 因为系统的使用者多是未受过专业训练的普通用户, 所以, 简单、自然、友好、灵活的图形界面必然受到他们的青睐. 而使用 ActiveX 控件来搭建组态软件的 GUI 就如搭积木一样方便、高效、便捷, 因此, 目前流行的组态软件多采用 ActiveX 控件作为其 GUI 的基本组成元素. 根据这一思想, 本文对组态软件 GUI 中常用的三个 ActiveX 控件进行了详细的分析和设计.

2 ActiveX 控件技术简介及开发工具的选择

ActiveX 控件技术是面向对象的开发技术. Ac-

tiveX 控件既是一个自动化对象, 也是一个标准的 COM 对象, 有些软件开发商致力于开发 ActiveX 控件, 通过 ActiveX 控件来快速地建立功能全面的应用系统.

ActiveX 控件的开发工具很多, 如 Borland 公司的 Delphi 7.0、微软公司的 VC++ 6.0、VB 等. 我们选择使用 VC++ 6.0 来设计 ActiveX 控件是因为 VC++ 6.0 捆绑的 MFC 技术使开发者可以不必关心具体的编程接口, 集中精力开发对象的动作就可以十分容易地完成控件的设计, 且控件的可移植性较好.

3 组态软件图形界面元素的 ActiveX 实现

在工业测试中, 经常要对远程测试现场的诸如电压、电流、气压、温度等特性进行测试与跟踪, 组态软件的 GUI 必须能够直观清晰地表达出这些信息, 以完成正确的人机交互, 因此, 界面友好、功能完备的 ActiveX 控件成为构建组态软件 GUI 的首选, 其中仪表控件、实时曲线显示控件以及滚动屏控件则发挥了更重要的交互作用.

3.1 ActiveX 控件设计的核心思想

VC++ 6.0 提供了 ActiveX 控件设计向导来快

收稿日期: 2004-09-02

作者简介: 张红斌(1979-), 男, 江苏如皋人, 助教, 研究方向: 多媒体技术、图像处理.

中国知网 <https://www.cnki.net>

速地完成 ActiveX 控件的设计. 向导为每个 ActiveX 控件都搭建了一个基本的框架, 设计者只需在这个框架的基础之上为控件添加外部属性、成员函数就可以了. 其中最重要的成员函数是控件类的 OnDraw 函数, 它是被系统时钟所控制, 每隔一个系统时钟该函数就对 ActiveX 控件的外型重画, 因此, ActiveX 控件的关键是编写 OnDraw 函数. 通过为控件编写不同的 OnDraw 函数就可实现不同的控件.

OnDraw 函数借助 VC++ 6.0 中提供的 GDI (Graphic Device Interface, 图形设备接口) 函数库. GDI 包含的绘图函数很多, 如画圆函数、画矩形函数等. OnDraw 函数的本质就是一组按特定流程组织的 GDI 函数的集合.

3.2 工业仪表控件的设计

3.2.1 工业仪表控件的分析

工业仪表控件的主要功能是根据现场采样的数据实时地通过仪表表盘显示出来, 并能根据数值的变化趋势(增或减)模拟指针的移动. 同时, 该仪表还可以设置测试数据的精度、范围等. 下图是工业仪表控件中的 OnDraw 函数的流程图.

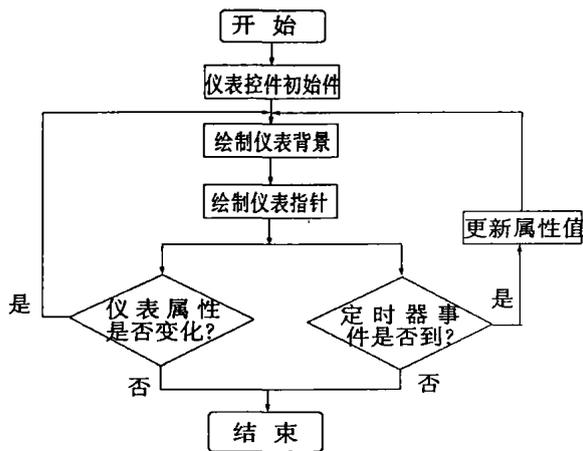


图 1 工业仪表控件 OnDraw 函数流程图

3.2.2 工业仪表控件的详细设计

根据图 1 的描述, OnDraw 函数的程序设计包括 4 个部分, 分别是绘制仪表背景、绘制仪表指针、绘制实时数据以及响应定时器事件, 其中仪表指针的绘制是整个设计的难点. 仪表指针是动态变化的, 例如上位机的测试电压是一个从 0V 至 100V 线性变化的, 要通过仪表控件来描述这样的过程关键就是动态变化的指针, 图 2 是指针形状示意.

指针实际是一个三角形, 它的顶点(图中 A)即指针的指尖, 它的其他两点(图中 B 和 C)是指针和表盘弧的交点. 指针绘制的流程是首先计算指针角

度, 若指针角度未超出表盘范围则计算指针坐标及两交点坐标并使用 GDI 函数完成指针的绘制, 否则矫正指针角度, 重新绘制指针. 计算指针指尖角度的方法是用终点角度减去当前值占量程的比例乘以满量程角度的值. 如果角度超出了最大值或最小值的范围, 就进行矫正, 核心程序如下:

```

brushFill.CreateSolidBrush(m-colorNeedle); // 创建画笔和刷子
penDraw.CreatePen(PS-SOLID, 1, m-colorNeedle);
;
pPenOld = pDC->SelectObject(&penDraw);
// 选择画笔和刷子
pBrushOld = pDC->SelectObject(&brushFill);
pDC->Polygon(pointNeedle, 3); // 绘制指针
画笔和画刷一起使用, 主要是因为要将画笔所围起来的区域用画刷填充颜色. 函数 Polygon 将 pointNeedle 数组的 3 个点按顺序连成一密封的区域. 这样画刷也能填充了.
    
```

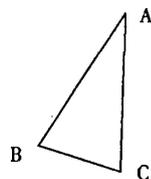


图 2 指针形状示意图

3.2.3 工业仪表控件的调试

控件在 VC 工具控件容器 ActiveX Test Container 中的调试如图 3 所示.

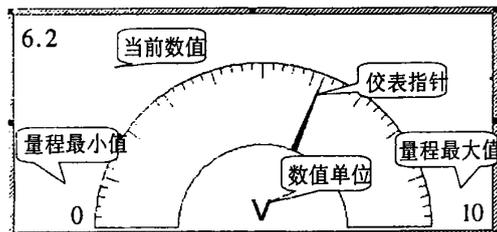


图 3 工业仪表控件调试图

3.3 实时曲线显示控件的设计

3.3.1 实时曲线显示控件的分析

实时曲线显示控件在工业测试及控制的应用十分广泛, 例如显示某电路模拟连续信号的实时变化. 在设计过程中还提供了 sin, cos, tan 三种数学函数与实时曲线进行对比. 该控件的整体流程和工业仪表控件类似, 它的 OnDraw 函数包括控件背景的绘制、坐标轴的绘制以及实时曲线的绘制. 控件属性较复杂, 表 1 列出了它的主要属性.

表1 实时测试曲线显示控件主要属性及功能对照表

属性名		功能说明
实时曲线取点数		用于设置显示区显示的点的个数
每格主刻度 表示的数值	水平方向	用于设置横轴的数值密度
	垂直方向	用于设置纵轴的数值密度
单位设置	水平方向	用于设置横轴的单位名称
	垂直方向	用于设置纵轴的单位名称
Y轴方向原点位置		用于设置原点在垂直方向上的位置
网格水平方向等分数		用于设置水平方向的等分数
网格垂直方向等分数		用于设置垂直方向的等分数

3.3.2 实时曲线显示控件的详细设计

控件设计的难点是如何实现曲线的动态平移显示. 曲线的平移能够反映现场数据的实时性, 以监测该点在现场工况变化情况下的控制稳定性, 因此曲线平移是实时曲线显示控件的一个最主要也是最重要的功能. 通常当前点在曲线的最右端, 随

时间推移, 整个实时曲线将动态地向左移动. 下面通过一组图来说明曲线平移的算法思想.

MFC 中 CBitmap 类封装了一个十分有用的位图操作函数 BitBlt, 它可将位图从一个设备场景复制到另一个设备场景, 实验表明这种以块的形式快速传递位图而不必重复绘制历史数据点, 可以实现曲线的平滑移动, 并且操作系统消耗资源少, 是一种值得推荐的好方法.

3.3.3 实时曲线显示控件的调试

3.4 滚动屏控件的设计

3.4.1 滚动屏控件分析

控件的功能是在屏幕上显示任意输入的字符, 字符可不断的向左缩进, 并且可以循环显示, 缩进的快慢也可调整. 设计的难点是如何循环显示所输入的字符串. 经过分析, 屏幕上字符串的移动可当作组成字符串的所有点的平移来处理, 将字符串存于一个动态数组中, 算术左移, 把第一个元素补在数组结尾, 再打印这个数组, 以循环移动字符串.

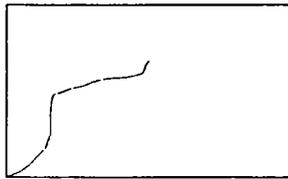


图 4-1 曲线不用平移

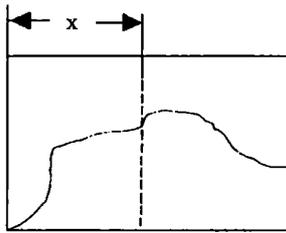


图 4-2 绘制到右边界, 以位图形式保存虚线右部, x 是曲每次平移的距离

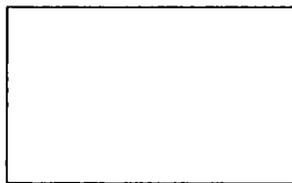


图 4-3 消影曲线, 等待重绘

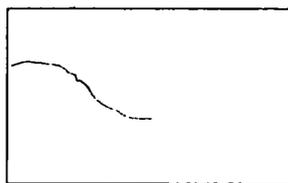


图 4-4 保存的曲线位图复制到屏幕的最左端

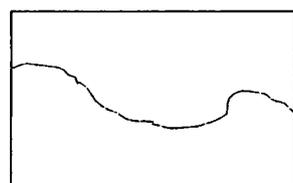


图 4-5 继续用最新的数据绘制曲线

3.4.2 滚动屏控件的详细设计

滚动屏控件也是通过 GDI 函数来绘制外观, 其中最重要的函数是绘制 Bezier (贝兹) 曲线的函数. Bezier 曲线的基本概念是使用四个点来设定任意一条曲线. (使用定点和控制把柄即两点构成直线, 两点在这条线上控制曲线形状)

在画一系列相连的 Bezier 曲线时, 只有当第一条 Bezier 曲线的第二个控制点、第一条 Bezier 曲线的终点 (也就是第二条曲线的起点) 和第二条 Bezier 曲线的第一个控制点线性相关时, 即这三个点在同

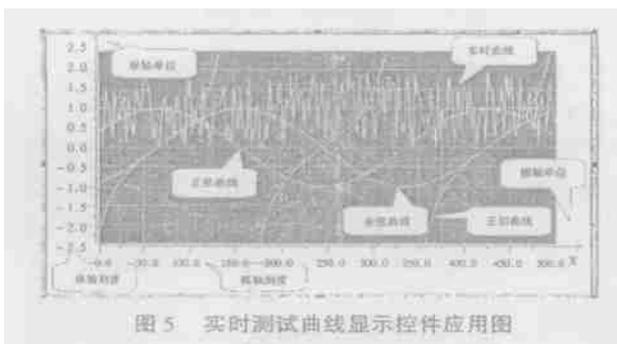


图 5 实时测试曲线显示控件应用图

一条直线时,曲线在连接点处才是光滑的,Bezier曲线并不通过事先指定的所有点,而我们有时希望圆滑曲线通过所有采样点.这时应该采用插值的办法来绘制圆滑曲线.Bezier曲线上各点坐标的插值公式是:

$$P(t) = \sum_{i=1}^n P_i B_{i,n}(t), t \in [0, 1]$$

其中, P_i 构成该 Bezier 曲线的特征多边形, $B_{i,n}(t)$ 是 n 次 Bernstein 基函数:

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$$

$$= \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, (i = 0, 1, \dots, n)$$

曲线绘制核心代码如下:

```
switch(bytes[i]) //判断当前点在贝兹曲线中的位置
```

```
{ case PT-MOVETO;
    pdc->MoveTo(points[i]); //移动当前点位置
```

```
    tStart = points[i]; break;
```

```
case PT-LINETO;
```

```
    pdc->LineTo(points[i]); break; //画直线
```

线

```
case PT-BEZIERTO;
```

```
    pdc->PolyBezierTo(points+i, 3); //画贝兹曲线,用于将当前画笔位置设为前一条曲线的终点
```

```
    i+=2; break;
```

```
case PT-BEZIERTO|PT-CLOSEFIGURE;
```

```
    points[i+2] = tStart; //画贝兹曲线并封闭图形
```

```
    pdc->PolyBezierTo(points+i, 3); //画贝兹曲线,用于将当前画笔位置设为前一条曲线的终点
```

```
    i+=2;
```

```
    break;
```

```
case PT-LINETO|PT-CLOSEFIGURE;
```

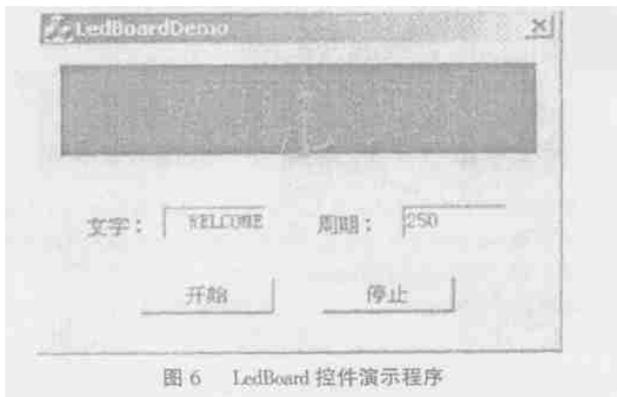
```
    pdc->LineTo(tStart); break;} //画直线并封闭图形
```

按顺序给出的前4个点分别是贝兹曲线的起

点、第一控制点、第二控制点和终点,此后每一条贝兹曲线只需给出3个点,因为后一条贝兹曲线的起点是前一条贝兹曲线的终点.

3.4.3 滚动屏控件的调试

控件调试如图6所示.



4 结束语

文中使用 VC++ 6.0 和 MFC 技术设计了三个 ActiveX 控件,对控件设计的一般流程进行了详细的介绍,运用此法,笔者还设计了其他6个 ActiveX 控件.使用这些控件并配合相应的矢量图形已经可以实现一个简单的组态软件的图形界面了,如果要扩充这个图形界面只要开发不同功能的 ActiveX 控件而不需要对开发环境的源代码进行修改,可见 ActiveX 控件技术的发展将会促进组态软件用户图形界面的不断改善.

参考文献:

- [1]王亚民,陈青,刘畅生,等.组态软件设计与开发[M].西安:西安电子科技大学出版社,2003.
- [2]陈志泊,王春玲.面向对象的程序设计语言—C++[M].北京:人民邮电出版社,2002.
- [3]林遂芳,冉宝春.基于ActiveX控件的网络虚拟仪器实验室[J].现代电子技术,2003,20:73-75.
- [4]侯俊杰.深入浅出MFC(第2版)[M].华中科技大学出版社,2001.
- [5]编写组.VC++6.0 MFC 类库参考手册[S].北京:人民邮电出版社,2002.

(下转第105页)

参考文献:

[1] 阮沈勇,等. MATLAB 程序设计[M]. 北京:电子工业出版社,2004.

[2] 程耀东. 机械振动学[M]. 杭州:浙江大学出版社,1994.

[3] 中川宪治. 工程振动学[M]. 上海:上海科技出版社,1981.

[4] 师汉民. 机械振动系统[M]. 武汉:华中科技大学出版社,2004.

Design and Implementation of Vibration-absorber Based on Matlab

QIU Ying

(School of Mechanical and Electronic Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: This thesis describes how to design a visual absorber system with the powerful calculation capability and GUI tool of MATLAB. This dynamic absorber design system can calculate the parameter of absorber and achieve a good vibration-absorbing effect according to the known dynamic parameters of the major system.

Key words: Matlab; GUI; vibration; absorber

(上接第 101 页)

Design of Basic Elements of Graphic User Interface of Configuration Software Based on ActiveX Controls Technology

ZHANG Hong-bin, LI Guang-li

Abstract: This paper presents the Graphic User Interface of Configuration Software, and introduces ActiveX Controls which compose of the Graphic User Interface. Finally it analyzes the design flow and key ideas of three ActiveX Controls in detail.

Key words: ActiveX Control; configuration software; VC++ 6.0; MFC